CET Dataset

In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [ ]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score,confusion_matrix, ConfusionMatrixDisplay
```

In [ ]:
```
df = pd.read_csv('data/CET_dataset.csv')
df.head(20)
```

Out[ ]:

	CET_score	Admitted
0	480	1
1	450	1
2	390	1
3	410	1
4	380	1
5	430	1
6	390	1
7	420	1
8	440	1
9	390	1
10	310	0
11	390	1
12	410	1
13	380	1
14	470	1
15	310	0
16	280	0
17	350	0
18	240	0
19	290	0

In [ ]:
```
df.shape
```

Out[ ]:
```
(160, 2)
```

In [ ]:
```
df.isnull().sum()
```

Out[ ]:
```
CET_score    0
Admitted     0
dtype: int64
```

In [ ]:
```
df.duplicated().any()
```

Out[ ]:
```
True
```

In [ ]:
```
df[df.duplicated()]
```

Out[ ]:

	CET_score	Admitted
6	390	1
9	390	1
11	390	1
12	410	1
13	380	1
...	...	...
155	350	0
156	370	1
157	280	0
158	290	0
159	390	1

94 rows × 2 columns

In [ ]:
```
data = df.values
```

In [ ]:
```
X = data[:,0:1]
y = data[:,1]
```

In [ ]:
```
X_train, X_test, y_train, y_test=train_test_split(X, y, test_size=0.2, random_state=213)
```

In [ ]:
```
X_train.shape
```

Out[ ]:
```
(128, 1)
```

In [ ]:
```
log_model=LogisticRegression()
```

In [ ]:
```
log_model.fit(X_train, y_train)
```

Out[ ]:
```
▾ LogisticRegression
LogisticRegression()
```

In [ ]:
```
pred=log_model.predict(X_test)
pred
```

Out[ ]:
```
array([0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       0, 1, 1, 0, 1, 1, 0, 1, 0, 1])
```

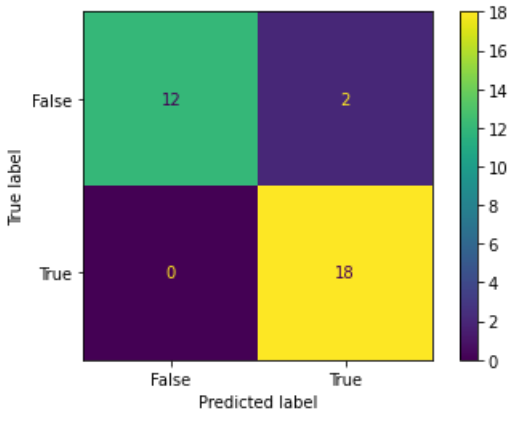
In [ ]:
```
accuracy_score(y_test, pred)
```

Out[ ]:
```
0.9375
```

In [ ]:
```
conf_matrix=confusion_matrix(y_test, pred)
conf_matrix
```

Out[ ]:
```
array([[12,  2],
       [ 0, 18]])
```

In [ ]:
```
cm_display1 = ConfusionMatrixDisplay(confusion_matrix=conf_matrix, display_labels=[False, True])
cm_display1.plot()
plt.show()
```



In [ ]:
```
print(log_model.predict(np.array([350]).reshape(-1,1)))
```

```
[0]
```

In [ ]:
```
print(log_model.predict(np.array([370]).reshape(-1,1)))
```

```
[1]
```